VSR: A Routing Protocol based on a Structure of Self-Organisation

Fabrice Theoleyre, Fabrice Valois

VSR: A Routing Protocol based on a Structure of Self-Organization

Fabrice Theoleyre* — **Fabrice Valois****

* *Grenoble Informatics Laboratory (LIG), UMR CNRS 5217*
681 rue de la passerelle, BP72, 38409 Saint Martin d'Heres Cedex, France

Fabrice.Theoleyre@imag.fr

** *Laboratoire CITI / INSA Lyon / INRIA*
21 av Jean Cappelle, 69621 Villeurbanne Cedex, France

Fabrice.Valois@insa-lyon.fr

ABSTRACT. *Mobile Ad Hoc Networks (MANets) are spontaneous wireless networks of mobile nodes without any fixed infrastructure. The routing problem focuses major attention since it suffers currently from a lack of performances and scalability. Besides, a self-organization structures the network and introduces a hierarchy. We propose here a new framework of routing protocols to benefit from a virtual topology of self-organization. The backbone is useful to introduce a first level of hierarchy and to limit the reactive overhead. Besides, clusters introduce a second level of hierarchy and create stable routes. VSR combines the assets of reactive and proactive flat routing with the self-organization hierarchy and allows local route repair.*

RÉSUMÉ. *Les réseaux mobiles ad hoc (MANets) sont des réseaux sans-fil spontanés de noeuds mobiles sans aucune infrastructure fixe. Le problème du routage a focalisé beaucoup d'attention car il souffre actuellement de performances médiocres et d'un manque de passage à l'échelle. Par ailleurs, une auto-organisation structure le réseau et introduit une hiérarchie. Nous proposons ici un framework de protocoles de routage bénéficiant de cette topologie virtuelle d'auot-organisation. La dorsale est utile pour introduire un premier niveau de hiérarchie et limiter le trafic de contrÙle reactif. Par ailleurs, les clusters introduisent un deuxième niveau de hiérarchie et créent des routes stables. VSR combine les avantages des approaches réactives et proactives avec la hiércrahie de l'auto-organisation et permet des réparations locales des routes.*

KEYWORDS: *virtual structure, self-organization, ad hoc networks, routing, hierarchy*

MOTS-CLÉS : *structure virtuelle, auto-organisation, réseaux ad hoc, routage, hiérarchie*

1. Introduction

Mobile Ad-hoc Networks (MANets) are literally networks *ready to work*. All terminals can communicate spontaneously with other nodes via wireless communications, without a fixed infrastructure. Furthermore, the network must function autonomously, organizing itself (Mario, 2005). Thus, the nodes should collaborate to attribute addresses, decide the radio frequency to use, exchange control traffic... Moreover, radio devices present a limited radio range. Hence, the source can be not directly connected to the destination. Intermediary nodes must forward the packet from the source to the destination: the network is multihops. Consequently, the nodes should collaborate in a distributed manner to exchange control information in order to set up routes: in MANets, a node is both client and router. Additionally, all nodes can move independently, creating sudden appearing and disappearing radio links, the topology changes continuously. Hence, the network must constantly adapt itself to the topology changes, maintaining routes, updating its knowledge about the network. MANets, because of their flexibility, are promised to a large spectrum of utilization. They could be useful for military operations, allowing radio connections from vehicles to soldiers without systematical satellite communications. MANets could be used for rescue operations after a earthquake having destroyed all telecommunications infrastructures. More generally, MANets could be deployed in any scenario of spontaneous information sharing (conference, classroom, home,...). Ad-hoc networks can sometimes be connected to the Internet via a dedicated device: the Access-Point (AP). Such ad-hoc networks constitute multihops cellular networks. In our opinion, they will constitute a natural evolution of peripheral networks. Such networks could be used by telecommunications operators to extend the coverage area of their classical cellular networks without any additional cost.

Ad hoc and hybrid networks constitute a large domain to study. All classical solutions must be reconceived because of some particular constraints. The radio utilization creates constraints about the bandwidth: the interferences reduce the capacity, the instability of radio links create sudden topology changes, a degraded reliability, and packet losses. Moreover, MANets are constituted by embedded terminals, presenting physical constraints: CPU, memory are limited. In the same way, energy is limited. A very efficient way to save energy is to cut off its radio device (Feeney *et al.,* 2001), entering in *sleeping mode*. We can also implement a topology control for example to reduce the power-energy consumption. Besides, the network is heterogeneous: laptops cohabit with PDA. The network must organize itself to balance efficiently the load among the nodes according to their capacity. Many topics must be addressed in MANets: a robust address assignment scheme, a solution securing data exchanges, an efficient interconnection of a hybrid network to the Internet, a mobility management scheme in the ad hoc area, etc. In particular, the routing problem is currently the most studied domain, since this represents a vital function in multihops networks. A data packet must be sent from a source to a destination optimizing the delay and the delivery ratio, but presenting a limited overhead. Traditionally, routing protocols are classified in *proactive* and *reactive*. In the proactive approach, a node knowns a priori all routes toward each node in the network. In the reactive propositions, a route is discovered on demand, only when it is necessary. However, whatever the protocol is, floodings are mainly used. Nevertheless, floodings present severe drawbacks of redundancy and reliability in MANets (Ni *et al.,* 1999). In consequence, new routing protocols must be proposed.

In our opinion, it is vital to first organize the network before any utilization. The network must present a hierarchy in order to improve the exchange of control packets. The self-organization structures represent currently a key point studied in MANets. As explained, the MANets are heterogeneous. Thus, a self-organization must take into account this heterogeneity in distributing different roles to each node, to create an organization. We proposed in (Theoleyre *et al.,* 2007) a virtual structure of self-organization constituted by a backbone and clusters. This virtual structure could be efficient to reach the following goals:

– scalability: because MANets are constituted by many nodes, clusters could group mobile nodes and backbone could concentrate flooding packets to minimize the broadcast storm problem.

– to hide some nodes neighborhood changes using a top-level view of MANets. This creates a stable and scalable view of the networks: the topology inside a cluster may change many times before a node exits the group. Furthermore, a protocol can limit the impact of the evolution of the topology at the cluster level.

– a virtual topology can elect *better* nodes as leaders and others as clients since it can take into account heterogeneous nodes.

A virtual structure of self-organization has already been proposed (Theoleyre *et al.,* 2007). It is now required to prove its efficiency to deploy easily efficient solutions for mobility management, addresses assignment…The contribution of this article is to present a framework of routing in MANets which uses efficiently this self-organization. Because of the virtual structure, we can achieve the following properties:

– stable routes are created because of the hierarchy

– the impact of floodings is minimized, and limits the overhead

– scalability is increased by combining several routing protocols which collaborate with each other to set up efficient routes

This proposition should be considered as a framework of routing protocols to benefit from a virtual topology of self-organization because any proactive and reactive protocols can be improved using this approach.

In the next section, a short overview of routing protocols and self-organization will be given. In the section 3 will be presented in more details the virtual structure used by the Virtual Structure Routing Protocol (VSR). Section 4 details the operations of VSR. Performances results are given in section 5. Finally, section 6 will conclude the article and will expose some perspectives.

2. Related Work

In this section, a short overview of the different self-organization structures will be given. Next, the different classes of routing protocols will be presented. We give here only a panorama since an exhaustive approach seems impossible because of the large number of propositions in this domain.

2.1. ***Structures of self-organization***

The main idea behind self-organization is to create a virtual topology which introduces a hierarchy in the network. These virtual structures allow to deploy routing protocols in an easier manner and more efficiently, as will be shown in the section 4. Such a self-organization is generally achieved through virtual backbones and clusters.

2.1.1. *Clusters*

To hierarchize the network to structure it is a promising approach. Clustering was already proposed, and is often based on an election process. In (Lin *et al.,* 1997), each node which has the lowest identity among all its neighbors which have no cluster, declares itself *clusterhead*. All its neighbors without any cluster join its cluster. Two types of clusters could be maintained. If no clusterhead is required ((Lin *et al.,* 1997)), the diameter constraint must be verified: each node can reach each other node in its cluster in at most 2 hops. A decision must be taken to choose the nodes to exclude, in order to optimize the number of new clusters. Oppositely, if the clusterhead is required, two maintenance sub-types could be achieved. If a clusterhead is maintained as long as possible, the cluster topology changes less frequently. A new clusterhead is elected only when some nodes become isolated. Oppositely, if the clusterhead is required to remain the node with the lowest identity of the cluster, more topology changes can occur.

(Amis *et al.,* 1999) proposes an algorithm to construct clusters with a flexible radius. To construct clusters of radius k, i.e. the maximal distance from one node to its clusterhead is k hops, $2k$ rounds are required. During the first k rounds, the lowest identities are propagated, k hops along. During the last k rounds, the highest identities of the kth round are propagated to force connected and non overlapping clusters. A node is elected clusterhead if it receives its identity during the lth round ($k + 1 \leq l \leq 2k$).

2.1.2. *Backbone*

Backbones are a key component of wired networks. So, many propositions were done to adapt this concept by constructing distributively a backbone in MANets (Alzoubi *et al.,* 2002, Butenko *et al.,* 2003, Wu *et al.,* 2003, Wu *et al.,* 1999). (Alzoubi *et al.,* 2002, Butenko *et al.,* 2003) proposes the construction of an approximation of a Minimum Connected Dominating Set (MCDS): each node is neighbor of at least one node of the MCDS, the MCDS is connected and of minimal cardinality. In a first step, some nodes are elected backbone members in a process similar to clustering. The second step consists in the interconnection of the clusterheads to form a backbone. (Butenko *et al.,* 2003) proposes an iterative exploration, presenting potentially a large delay. (Alzoubi *et al.,* 2002) presents a *best-effort* approach optimizing the delay to the detriment of the cardinality. (Wu *et al.,* 1999, Wu *et al.,* 2003) proposes a localized algorithm: each node which has a connected set of neighbors with a lower identity and which covers all its own neighborhood is not in the Connected Dominating Set (CDS). (Theoleyre *et al.,* 2007) proposes the construction of a structure combining a backbone and clusters, with a flexible distance from one node to the backbone. More details will be given in the next section.

2.2. *Flat routing*

Several routing protocols have been proposed for MANets. In the flat approach, all nodes are considered equal. Three main classes exist: proactive, reactive and hybrid.

2.2.1. *Proactive*

In the proactive approach, each node maintains a route toward each other node in the network. However, if each node sends periodically topology packets in the network, the medium is heavily loaded and a broadcast storm will surely occur. Thus, the goal of the proactive protocols is to limit this overhead.

DSDV (Perkins, 1994) uses the Bellman-Ford algorithm: the topology packets are only exchanged with the neighbors. If a node receives a route toward an unknown destination or a shorter route toward a known destination, it updates its routing table. In the next advertisement, the new routing table will be sent. A sequence number avoids loops in routes. However, the convergence delay could be significant. OLSR (Clausen *et al.,* 2003) is a link-state routing protocol and proposes to limit the impact of the flooding. Each node selects a subset of its neighbors, the *Multi Point Relays* (MPR), to cover all its 2-neighborhood. Further, when a topology packet is transmitted, only the MPR will forward it. Recursively, only the MPR of the MPR will forward the packet. In consequence, the overhead is greatly reduced. However, a node will in many scenarios communicate with a few destinations, although it maintains all the routes. Moreover, a topology packet for each potential destination must be received by each node in the network. In consequence, the overhead remains large.

2.2.2. *Reactive*

Reactive protocols create routes *on demand*. When a node has a data packet to send and no route is available in its routing table, it initiates a route discovering. In DSR (Johnson *et al.,* 2004), a node sends a `Route Request` (`RREQ`) in broadcast. The `RREQ` is forwarded and accumulates the addresses of intermediary nodes. When the destination receives it, it answers with a `Route Reply` (`RREP`), sent along the inverse route contained in the header of the `RREQ`. The source can finally add the route in its routing cache. When the route is broken, a new route discovering is initiated.

Reactive protocols allow to limit the memory required for the routing table and are efficient when a node has only a limited set of destinations. However, the latency before the route set up could be problematic for some applications. Additionally, the multiplication of `RREQ` flooded in the whole network can create a broadcast storm. Finally, a route is not maintained: it is reconstructed when it becomes broken. In AODV (Perkins *et al.,* 2003), routing caches are distributed in the network. When a `RREQ` is forwarded, an entry in the routing cache pointing to the source is added. When a `RREP` arrives, it is forwarded according to the routes in the cache. Moreover, any node which has an entry for the searched node can answer to a `RREQ`. The packet length is reduced because no route is contained in the header. However, routing caches can present inconsistencies because of mobility: a node moved and the route became broken. If the routing cache of the previous hops was not flushed, a stale route will be used. The timeouts in the cache must be a trade-off between the overhead generated per the route requests and the node's mobility.

2.2.3. *Hybrid*

ZRP (Haas *et al.,* 1998) proposes a trade-off combining the reactive and proactive approaches. Each node maintains a proactive route toward each node at most p hops far, constituting its zone, p being the zone radius. The inter-zone routing is reactive. A node in unicast sends the `Route Request` to each node exactly p hops far, its *border nodes*. These border nodes forward the `Route Request` to their own border nodes. Finally, the route is contained in the header and comprises all the border nodes addresses. Routing from one border node to another is possible thanks to the proactive intra-zone routing. However, the zones are source oriented. Thus, zones can overlap, and create a flooding worse than the blind flooding when for example a node forwards several times a `RREQ` to different border nodes. Thus, (Pearlman *et al.,* 1999) proposes to select adequate border nodes, but according to the (2p+1)-neighborhood knowledge, which creates a significant proactive overhead. Moreover, zones are not defined hierarchically. Thus, a more sophisticated structure which reflects the natural heterogeneity of the network could be more efficient. (Helmy, 2005) extends ZRP in proposing a method to discover reactively border nodes and maintain their list in order to reduce the overhead.

2.3. *Clustering Routing*

CBRP (Jiang *et al.,* 1999) is a routing protocol based on clustering. Each node sends periodically `hello` containing the list of neighbors, and the list of adjacent clusters. An adjacent cluster of the node N is a cluster for which one of its members is neighbor of N. By extension, the adjacent clusters of the cluster C is the union of the adjacent clusters of the nodes in C. Each node knows its 2-neighborhood, and clusterheads know additionally the clusters adjacent to their own cluster, and the *gateway* nodes to reach them. CBRP is a source routing protocol, the route being always contained in the header of the packet. When a node has a data packet to

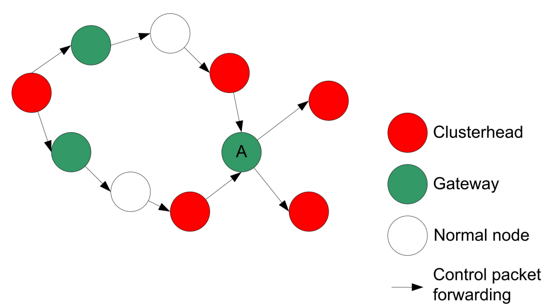

Figure 1. *Suboptimal case for the flooding behavior in CBRP*

send, its clusterhead acts as a proxy. When a clusterhead receives a `RREQ`, it sends the request to its cluster. Then, if the destination is not present in its cluster, the clusterhead forwards the `RREQ` to all adjacent clusters. A gateway is chosen to reach in unicast each of these clusters. The destination will send a `RREP` in unicast. Each clusterhead tries to optimize locally the route and then forwards the `RREP`. However, the route is constituted by a list of nodes identities: CBRP does not benefit entirely from the hierachized topology. In particular, one topology change could be sufficient to break a route. Besides, CBRP proposes a route repair mechanism. When a node does not receive an acknowledgment from the next hop, it tries to reach it through one intermediary neighbor, chosen according to its neighborhood table. The route is in consequence lengthen by one hop for each repair. Finally, the flooding of `RREQ` is suboptimal since the topology formed by the clusterheads and gateways forms many loops. In some cases, the CBRP flooding could be worse than the blind flooding (fig. 1: the gateway A is chosen twice during the flooding, for two different clusterheads).

2.4. *Backbone Routing*

CEDAR (Sinha *et al.,* 1999, Sivakumar *et al.,* 1998) proposes to create a spine of clusterheads interconnected by virtual links. Clusterheads send periodically `hello` 3 hops far to establish virtual links with adjacent clusterheads. The overhead could be large. Moreover, the virtual links can become quickly sub-optimal since the environment is dynamic. A more stable backbone and a more complex hierarchy could improve the routing performances. In (Sivakumar *et al.,* 1998), the backbone is only used for topology packets flooding for a link state routing protocol, which is in our opinion a restricted utilization of a backbone. In (Sinha *et al.,* 1999), each node advertises the bandwidth changes of all its links. These changes are flooded through the backbone. A mechanism based on differentiated queues allows to propagate only stable links far in the network. Then, the backbone is in charge to compute routes, not passing through the backbone. The backbone is only used for backup routes. However, a link-state mechanism presents a significant overhead: hybrid protocols combining the assets of proactive and reactive could be more efficient. Moreover, the backbone is not as stable as it could be.

2.5. *Performance comparison*

(Boukerche, 2004) proposes a performance evaluation comparing several protocols (AODV, DSDV, DSR and CBRP). AODV is shown to outperform CBRP for the delay. However, AODV presents a lower throughput than CBRP. In our simulations, CBRP presents lower performances since the degree is smaller, and CBRP is less efficient in sparser networks. Finally, DSDV presents a low overhead but significant packet losses since it is a table-driven protocol and problems of convergence appear when many topology changes occur. Instead of comparing reactive and proactive approaches, we focus here in the proposition of a self-organization structured routing protocol, and the comparison of its performances with classical flat and hierarchical approaches.

3. A Virtual Structure of Self-Organization

3.1. *Motivations*

A virtual structure helps in our opinion to structure the network and to organize it. After organizing the network, a routing protocol could be easier to deploy, and more importantly, more efficient. The virtual structure described in (Theoleyre *et al.,* 2007) combines the assets of a backbone and clusters.

All construction and maintenance algorithms were proved to be self-stabilizing in (Theoleyre *et al.,* 2005): whatever the state of the network is, a valid self-organization structure is obtained after a finite time. Moreover, simulations proved that the algorithms converge after a few seconds, even if a network of 100 nodes starts from scratch (i.e. all the nodes arrive in the network simultaneously).

We will present here shortly this self-organization structure, in order to clarify the routing algorithm description of section 4.

3.2. *Neighborhood Discovering*

The algorithms for both construction and maintenance require the k_{cds}-neighborhood knowledge (k_{cds} being a parameter of the protocol). To reach such a goal, each node sends in broadcast `hello` packets containing the list of its neighbors. These `hello` are propagated k_{cds}-1 hops far. A node will forward the `hello` if it comes from a bidirectional neighbor. Thus, we avoid the creation and utilization of unidirectional links.

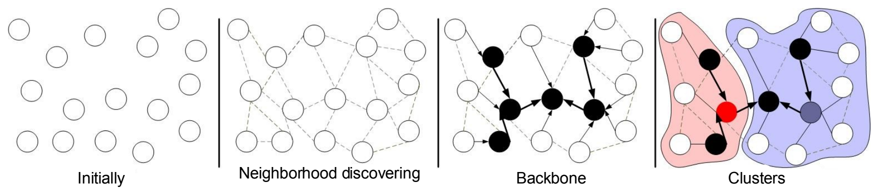

Figure 2. *ProcÈdure de construction de la structure d'auto-organisation (*$k_{cds} = 1, k_{cluster} = 2$*)*

3.3. *The self-organization structure*

The self-organization consists in a backbone and a cluster structure. A backbone is useful to collect the control traffic and optimize the information dissemination. Moreover, a hierarchy is created among the backbone clients and the backbone members. The backbone constitutes a natural extension of the backbones of classical wired networks. Besides, clusters hierarchize the network in creating services areas, the clusterhead being the manager of its area. It offers a global simple and stable view of the radio topology: a node can move inside its cluster, it has no impact on the cluster topology view. Some radio topology changes are hidden. To construct stable topologies, the backbone nodes and clusterheads are elected according to a stability metric combining several criteria (relative mobility, energy and degree). A performance evaluation with simulations demonstrated the low number of clusterheads and backbone nodes, the robustness to mobility and the stability of this self-organization in (Theoleyre *et al.,* 2007).

3.3.1. *Backbone*

The backbone is constructed before the clusters. Thus, the distance via the backbone from a node to its clusterhead can be limited, only backbone nodes can participate to the clusterhead election (optimizing the overhead) and a clusterhead is forced to be a backbone member. Moreover, the maximum distance from one node to the backbone (k_{cds}) is a parameter of our solution. In volatile environments, a small k_{cds} allow to limit the backbone disconnections. In quasi-static environments, k$_{cds}$ could be high since less topology changes occur.

A node can be either dominator (backbone node), dominatee (at most k$_{cds}$ hops far from the backbone), active (in election) or idle (the node is in erroneous or initial state). A leader triggers the construction by becoming dominator. If the ad hoc network is connected to the Internet, the Access Point is the natural leader. If several AP exist, the self-organization consists just in one backbone per AP, interconnected by the wired links. Thus, the set of backbone radio links and wired links form a common global backbone.

An election process allows to create in a first time a *dominating set*: any node is at most k_{cds} hops far from at least one backbone node. Then, the dominators are interconnected so that the backbone forms a connected structure. Moreover, the backbone consists in a tree and the leader is the root of this tree. Each node maintains a parent, one hop nearer from the root.

Since the radio topology changes, the backbone must be adapted. We proposed algorithms to maintain the backbone structure. In particular, some special control packets, the `leader-hellos`, are flooded in the backbone, forwarded only by dominators. These packets allows to detect backbone disconnections. We proposed local procedures to reconnect the backbone, to maintain a connected structure. In the same way, other procedures allow to eliminate the backbone redundancy: a backbone node should become dominatee when it has no dominatee exactly k_{cds} hops far and when it is not necessary for the backbone connection. Consequently, we limit the backbone cardinality, which is useful for example to limit the overhead of a backbone flooding. The algorithms maintain the tree structure, which could be useful to discover for example a route to the Internet through the Access Point.

3.3.2. *Clusters*

As explained above, only dominators take part in the construction of clusters: a dominatee joins automatically the cluster of its parent.

During the construction phase, each dominator sends special `cluster-hello` packets $k_{cluster} - k_{cds}$ hops far, forwarded only through backbone links. A dominator becomes clusterhead when it has the highest weight (the stability metric value) among all other dominators at most $k_{cluster} - k_{cds}$ hops far and without clusterhead. When a node becomes clusterhead, it sends immediately a `cluster-hello` so that its neighbors can choose it as clusterhead. These special `cluster-hellos` are only required during the construction phase.

We also proposed maintenance algorithms. In `Hello` packets are inserted additional information: the clusterhead id and its distance in hops via the backbone links. Thus, each node can implement a distance vector algorithm to maintain a valid clusterhead, at most $k_{cluster} - k_{cds}$ hops far. In the same way, redundancy elimination procedures allow to limit the number of clusterheads.

4. Virtual Structure Routing

4.1. *General Description*

We focus here on the problem of routing in MANet: a data packet must be delivered with a minimal delay and without any loss to a destination. Our proposition takes into account some key properties in MANet: the network is dynamic, some packet losses can occur and the network is naturally heterogeneous. Moreover, the routing protocol must be scalable with both the network cardinality and the traffic load. We use to reach this goal the virtual structure of self organization described in the previous section. The backbone is useful to optimize the overhead by controlling the impact of the flooding. The clusters help to create a simple and stable virtual view of the topology, to create stable routes.

We explain here how to build an hybrid protocol based on the virtual structure previously described. This hybrid protocol combines the proactive and reactive assets. The delivery ratio is maximized since routes are updated continuously: a local proactive protocol in each cluster updates continuously the knowledge of the local topology. On the contrary, the overhead is minimized and routes are stable: a reactive protocol is used for long routes, and the stability is improved since a route is constituted by a list of clusters (instead of a list of nodes).

4.2. *Intra-Cluster Routing*

We propose to deploy a proactive routing protocol for routes inside the clusters. We can assume a local traffic pattern: nodes will mainly exchange data packets with nodes in proximity, and sometimes with farther nodes. Such a traffic pattern is scalable with the network capacity (Li *et al.,* 2001). Additionally, even if such a traffic pattern is not relevant, the local proactive routing protocol allows to optimize locally in a cluster the long routes: the route is only constituted by a list of clusters, and the local route inside each cluster is computed with the recentest information, thanks to the proactive part of the protocol. Consequently, each node must know the topology of its cluster. More precisely, we will see a few lines below that the topology knowledge of the k$_{cluster}$-neighborhood of the cluster is sufficient.

We chose to fix $k_{cds} = 2$, this value being a good trade-off: only 25% of the nodes are backbone members with an average degree of 10 (Theoleyre *et al.,* 2007). For the backbone maintenance, the k$_{cds}$-neighborhood knowledge is required. Thus, `hello` packets are periodically broadcasted. Since a node must distinguish unidirectional and bidirectional links, each node must insert the list of its 1-neighbors in the `hellos`. In conclusion, a node just broadcasts periodically `hellos` so that each node can reconstruct the topology of its 2-neighborhood.

Since the knowledge of the $k_{cluster}$-neighborhood is required, this scheme is not sufficient. Thus, a node must additionally forward an `hello` if it comes from a bidirectional neighbor of its cluster, the initial TTL being set to $k_{cluster}$. In consequence, each node knows all the nodes in its cluster at most $k_{cluster}$ hops far. Additionally, the backbone is well maintained since a node keeps on receiving `hellos` in broadcast from neighbors not in its cluster.

Each node computes optimal routes with flexible criteria (hops, link quality...) when it has collected the radio links of its k$_{cluster}$-neighbors. For example, the Dijkstra algorithm can be used. The default route is pointing to the clusterhead. When a node must forward a packet inside its cluster, the following rules are applied:

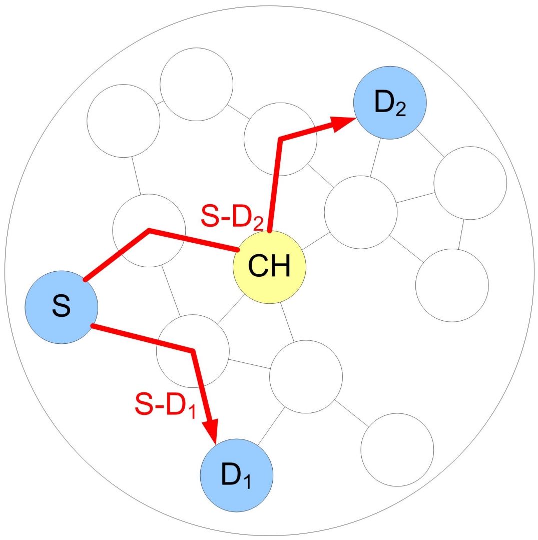

Figure 3. *Intra-cluster routes in a cluster of radius 2 with the 2-neighborhood knowledge*

– If a route to the destination is known, the packet is forwarded to the next hop

– If no route is known, the packet is sent via the default route, i.e. toward the clusterhead. Since each node is at most $k_{cluster}$ hops far from its clusterhead, in the worst case the packet will reach the clusterhead. Moreover, the clusterhead knows the topology of the whole cluster since each node in the cluster is by construction at most $k_{cluster}$ hops far. However, such a case occurs seldom, only when the source and the destination are in opposite sides of the cluster. Thus, the shortest route in this configuration will with high probability pass through the clusterhead (cf. fig 3). Consequently, this does not constitute a severe drawback.

4.3. *Inter-Cluster Routing*

Longest routes will be discovered on demand and will be constituted by a list of clusters instead of a list of nodes. The cluster topology being more stable, the stability of the discovered routes will be improved. Additionally, the proactive routing protocol will allow to route efficiently inside a cluster.

4.3.1. *Topology Discovery*

A reactive route being a list of clusters, an intermediary node must forward a packet to another node in its cluster which will forward it to the next cluster. The identifier of a cluster is chosen to be the identifier of its clusterhead. Such an addressing scheme presents an obvious interest of simplicity. The local routes are already known with the proactive intra-cluster routing protocol. However, a mechanism to discover adjacent clusters must be proposed. We propose to integrate this function in the neighborhood discovery. A node which receives `hellos` from neighbors in a different cluster can act as a gateway for this cluster: it will advertise in its `hellos` the identities of all adjacent clusters. Since an `hello` is forwarded k$_{cluster}$ hops far, some nodes can have no gateway toward a specific cluster: the default route through the clusterhead will be used. The clusterhead will surely know the route.

4.3.2. *Route Discovery*

When a node S wants to send or forward a unicast packet, the following possibilities can occur:

– the destination D is at most k$_{cds}$ hops far and in a different cluster, or D is in the same cluster and at most k$_{cluster}$ hops far. Thus, S executes the proactive intra-cluster routing protocol

– a route toward D is present in the routing table. Thus, S executes the inter-cluster routing protocol

– D is unknown, S will initiate a route discovery. One can notice that if D is in the same cluster as D but strictly more than $k_{cluster}$ hops far, a route discovery will be initiated although D could be reached by the intra-cluster routing protocol. However, we avoid with such a mechanism to centralize in the clusterhead the route discoveries and thus the traffic: routes will be discovered by the source and entries will be added in the routing table of the source. The clusterheads will not represent the bottlenecks of the network.

However, the overhead of a route discovery must be minimized. We propose to use the backbone to achieve this goal. The source S generates a `RREQ`. If S is a dominatee, it sends the `RREQ` to its parent in the backbone. The first dominator will add its cluster in the route of clusters id contained in the header of the packet. Then, it sends in multicast the `RREQ`, initiating a backbone flooding. A dominator Dom which receives a `RREQ` lookups in its intra-cluster routing table if the searched destination D is present:

– If D is unknown, Dom appends its clusterhead if it is not present in the cluster route contained in the packet. Then Dom forwards the request in multicast to other backbone members

– If D is present, Dom appends its clusterhead and the clusterhead of D in the route if they are not present. Then, it creates a `Route Reply` and copies the cluster route present in the `RREQ` after inversing it. Finally, Dom sends the `RREP` in unicast, executing the inter-cluster routing algorithm. Dom acts as a proxy for the `RREP`, saving on average k_{cds} radio transmissions (to and from the destination, on average $\frac{k_{cds}}{2}$ hops in each direction)

In VSR, contrary to DSR or AODV, the `RREQ` is only forwarded by the backbone nodes, saving many useless transmissions. Moreover, the backbone being a tree, a backbone member which generates a `RRREP` will stop the backbone flooding for the whole branch: in a classical approach, the flooding can often not be stopped, even if the destination is found.

We choose source routing for the inter-cluster routing: if the route of clusters is stored in the nodes of the route, the hop-by-hop route could not be changed on the fly, else the nodes of the new route would not have the cluster route in their routing table.

4.3.3. *Routing*

A `Data` packet or a `RREP` is sent in unicast and contains in the header the route of clusters from the source to the destination. Before relaying a `Route Reply`, a node may add in its routing cache the route to the source and destination, reducing the number of further `RREQ`. When a node N_1 (in cluster C_1) receives a packet to forward, it executes the intra-cluster routing algorithm. If a destination is found, the packet is directly forwarded to the next hop. Else, N_1 searches the first known cluster C_2, nearest of the destination. It executes the rules in the following order:

1) A 1-neighbor N_2 is in the cluster C_2: N_2 constitutes the next hop

2) A 1-neighbor N_2 is in the cluster C_3 and is gateway for the cluster C_2. N_2 constitutes the next hop

3) A node in the cluster C_1 is gateway for C_2 in the neighborhood table of N_1 . N_1 chooses the nearest gateway if several exist. Then, the intra-cluster routing is executed to find the next hop N_2 toward this gateway. The packet is forwarded to N_2

4) Else, N_1 forwards the packet to the next hop toward its clusterhead. Such a case can occur since N_1 do not know the gateways in its cluster strictly more than $\mathrm{k}_{cluster}$ hops far. However, if the packet reaches the clusterhead, the clusterhead has a complete knowledge of its cluster since any node of its cluster is by construction $\mathrm{k}_{cluster}$ hops far. Thus, it knows at least one gateway for the next cluster if one exists. If no gateway is known, the clusterhead will drop the packet since an inconsistency appeared in the routing table. It will additionally send a `Route Error` to the source.

To avoid routing loops, a node can forward the packet to a cluster C_{next} only if C_{next} is in the route of clusters nearer from the destination than its own cluster. Thus, the packet will always be forwarded one hop nearer from the destination.

However, inconsistencies in the neighborhood tables can appear if the nodes are mobile (Wu *et al.,* 2004): the `hello` packets are sent only periodically. If a topology change occurs between two transmissions, the local view of the topology is erroneous. Moreover, packets suffer from transmission delays and collisions. In consequence, to avoid routing loops, a packet is silently dropped if it has already been received before. Duplicate packets are detected thanks to their id.

The route is dynamically computed hop by hop. Additionally, the cluster route could be shorten on the fly since a node chooses the adjacent cluster nearest of the destination. When a node forwards a `Route Reply` and updates the cluster route, it updates the route in the packet header. The destination which will receive the `Route Reply` will cache a valid and shorter route of clusters.

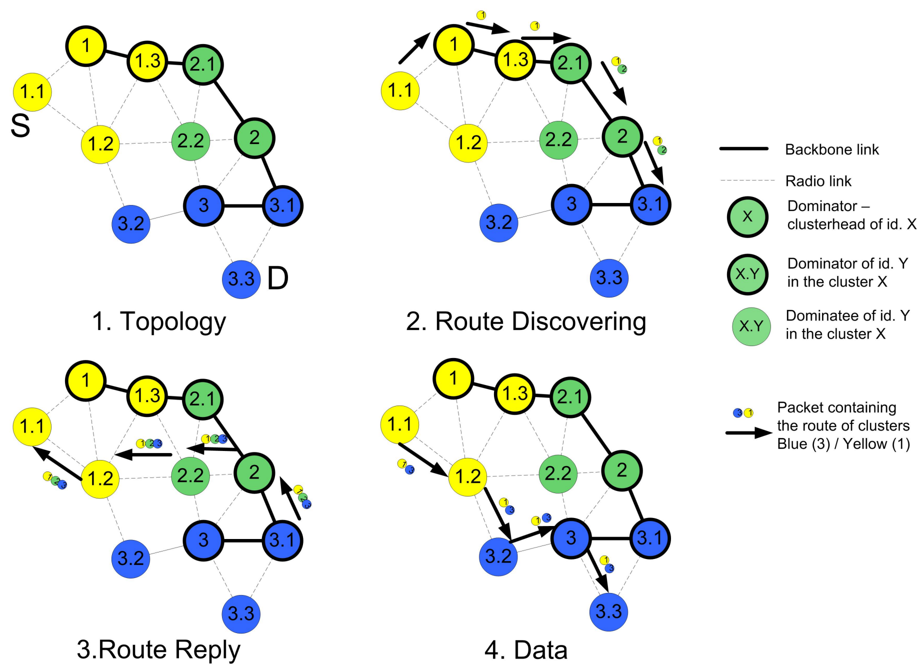

Figure 4. *Example of a network topology*

Finally, the route is not the shortest route, but we try to reduce its length by choosing to forward the packet to the cluster nearest of the destination. Consequently, routes are almost shortest routes, as corroborate the simulations in section 5.

The inter-cluster routing is robust: a list of clusters to follow constitutes the route. The hop by hop route is computed dynamically on the fly, with the recentest local knowledge, convergence delays being smaller when the searched node is nearer. If the cluster route remains valid, data packets and `Route Replies` will reach the destination even if many individual nodes move inside their cluster. Additionally, because the cluster topology is more stable than the radio topology, the routes are also more stable.

4.3.4. *Example*

Let the graph represented in figure 4 be the network topology. The source 1.1 wants to send a data packet to the destination 3.3.

Node 1.1 does not know any route toward node 3.3. It buffers the data packet and sends a `Route Request` in unicast to its dominator, node 1. The packet does not contain any route of clusters. Node 1 receives the packet and adds its clusterhead in the route. Then, it forwards the `Route Request` in multicast to the backbone members. Node 1.3 forwards the request. Node 2.1 adds its clusterhead in the route and forwards the packet. Finally, node 3.1 receives the `Route Request`. Node 3.3, the destination is neighbor of node 3.1 and is in consequence in its neighborhood table. Thus, node 3.1 generates a `Route Reply` with the route *3/2/1*.

Node 3.1 tries to reach the cluster 1. However, the gateway to cluster 1 in its cluster is more than $k_{cluster}$ hops far ($k_{cluster} = 1$), and is unknown from node 3.1. However, node 3.1 has a neighbor 2 which is itself in the cluster 2. Node 3.1 sends the `Route Reply` in unicast toward this node. Node 2 receives the `Route Reply`. It knows the gateway 2.2 for the cluster 1. Node 2.2 forwards the packet directly to the cluster 1, and finally the packet reaches node 1.1 thanks to the intra-cluster routing protocol.

Finally, node 1.1 receives the route of clusters *1/2/3* to reach node 3.3. Node 1.1 adds the route in its cache and sends the data packet after updating the route of clusters in the header. Node 1 tries to reach the cluster 3: it finds the gateway 1.2. Thus, node 1.1 updates the route in its routing table and in the header of the packet. Node 1.2 forwards the data packet to node 3.2. Node 3.2 does no know the destination 3.3 (more than k_{cds} hops far) and forwards the packet to its clusterhead. Finally, the data packet is received by node 3.3.

4.4. *Route Maintenance*

We assume that the delivery ratio represents a key metric of efficiency for routing protocols. Thus, we propose a very simple mechanism for route repairs. Several methods of acknowledgment exist:

– MAC acknowledgment: if the MAC layer fails to send a packet in unicast, a notification is sent to the higher layer. No overhead is required. This method is not currently feasible for many wireless network cards because of problems of implementation.

– Passive acknowledgment: each node N is in promiscuous mode and verifies that the next hop forwards the packet. If N does not hear any forwarding after a timeout, N retransmits the packet. If for any reason the next hop has already forwarded the packet, and N did not hear it, the next hop will explicitly send an `Acknowledgment packet`. No overhead is required, except for the final destination which must send an explicit ack since it does not forward the packet.

– Active acknowledgment: when a node receives a packet, it sends automatically an explicit packet to acknowledge the previous hop. A large overhead is created, perturbating the radio medium.

We assume that such an acknowledgment mechanism is implemented. If a node fails to receive an acknowledgment for the next hop, it initiates a route repair. It just re-execute the routing algorithm, but the erroneous node is considered dead and is forbidden as a next hop. This route repair limits the impact of the convergence delays in the neighborhood table after a topology change. It largely increases the delivery ratio as can be seen through simulations in section 5.

When a node fails to repair the route or tried max unsuccessful repairs, the route is considered as broken. A `Route Error` is generated and forwarded in unicast along the inverse route, using the inter and intra-cluster routing protocols. All the intermediary nodes must update their routing cache for the failed destination. The source will finally receive this `Route Error`, buffer the next data packets, and generate a new route discovery.

5. Performance Evaluation

We present here results about simulations using OPNET Modeler (OPNET Modeler, n.d.). We used the 802.11 model proposed in OPNET with a standard 300m radio range, in DCF mode, without RTS/CTS. Each node moves itself following the random waypoint mobility model, without pause time. All results are computed with a 95% confidence interval. We consider as general values a mobility of 5m.s^{-1}, 60 nodes, a degree of 10, and 4 simultaneous flows. An average degree of 10 nodes could be easily achievable with a topology control algorithm like (Li *et al.,* 2005). Topology control allows to adapt the radio range, save energy, and decrease the degree limiting interferences. In all ways, the impact of the degree is evaluated in the section 5.4.

We do not present here the behavior of the self-organization algorithms since they were deeply studied in (Theoleyre *et al.,* 2007, Theoleyre *et al.,* 2005).

The traffic generation is modeled as follows: flows of 20 `data packets` interspaced by 0.25 s are sent. For each flow, a destination and a source are randomly chosen. We did not choose a local traffic pattern (Li *et al.,* 2001) in order to obtain the most general case as possible. The inter-flow time follows an exponential distribution centered on 5 seconds to keep a constant average number of simultaneous flows. The packet size follows an exponential distribution centered on 128 bytes.

The results detail the relevance of this routing framework based on a self-organization. Particularly, the horizontal and vertical scalabilities (i.e. impact of the number of nodes) and the impact of mobility, density and the overhead are studied. We compare the performances of VSR with the performances under the same conditions as AODV and OLSR. To evaluate the impact of the properties of the virtual structure, we chose to simulate VSR with $k_{cds} = 1/k_{cluster} = 2$ and with $k_{cds} = 2/k_{cluster} = 3$. To have a fair comparison among the protocols, the retransmission mechanism was deactivated for each protocol if one exists. Finally, separated simulations present the performances gain when acknowledgments and route repairs are implemented. To have the most common scenario, passive acknowledgments are considered not available, and active acknowledgments were implemented.

5.1. *Horizontal Scalability*

We investigate firstly the horizontal scalability of the different routing protocols (fig. 5), i.e. the impact of the number of nodes. The end-to-end delay of AODV

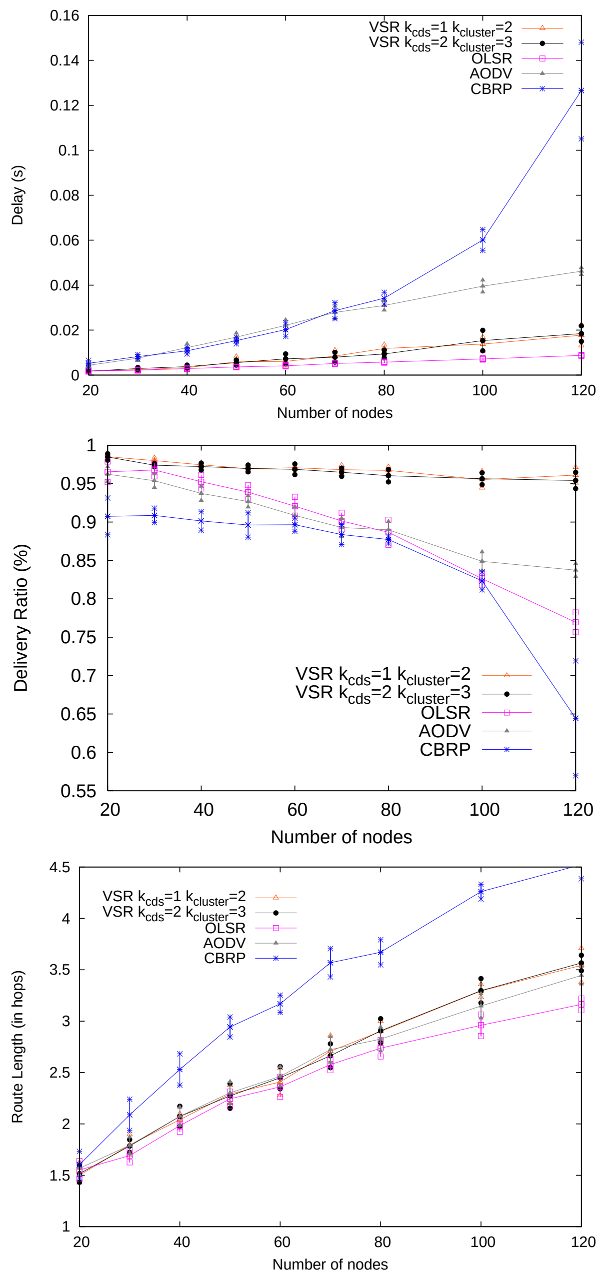

Figure 5. *Horizontal scalability*

and CBRP is higher and increases when the number of nodes increases: the protocols being reactive, more nodes must forward the `RREQ`. Thus, the time for the route discovering increases, which impacts the global delay. The delay of OLSR is minimal since OLSR is a proactive protocol: a route is immediately operational, and no delay is required. VSR, whatever the backbone radius is, presents a stable delay, near from the delay of OLSR: the backbone structure helps to optimize the route discovering. Furthermore, the clustered structure allow to maintain a hierarchy in the network without creating an additional delay.

We studied the length of the routes created by the protocols (fig. 5). CBRP presents the highest route length: the route discovering following the cluster topology, the length could be increased. The mechanism of self-deletion in the route header when a clusterhead forwards the `RREP` is not sufficient to have the shortest routes. AODV is a reactive protocol but the average route length is lower than CBRP and similar to other protocols. OLSR discovers always the shortest routes since it is a link-state routing protocol. The route length of VSR seems independent from the parameters $k_{cluster}$ and k_{cds}. Furthermore, the routes seem very near from the shortest routes of OLSR: the mechanism of forwarding to the nearest known cluster, and the local proactive knowledge of the cluster seem efficient to propose short routes. The cluster hierarchy can be easily exploited without any route lengthening.

These remarks are corroborated by the study of the route length distribution (fig. 6): the proportion of the routes which are exactly x hops long are reported on the graph for each protocol (x varying from 1 to 9). OLSR and VSR (whatever the backbone radius is) present a very similar distribution: both protocols achieve to discover shortest routes. AODV tends to discover longer routes, but the distribution is relatively near from the distribution of OLSR. CBRP discovers the longest routes. Whereas less than 3% of the routes of OLSR are more than 7 hops long, 17,5% of the routes of CBRP are longer than 7 hops. A longer route will increase the probability of collisions with other `RREQ`, `RREP` or `Data packets`. Thus, the route discovering will be repeated several times to have a reply, increasing the load on the medium. In the same way, some `Data packets` will be lost.

Then, we measured the delivery ratio, i.e. the ratio of `Data packets` which are finally well-delivered to the destination. CBRP seems to suffer from its sub-optimal route discovering: routes are longer, and more collisions occur. Thus, the packet losses are more significant. The delivery ratio of AODV is higher but the route discovering creates many collisions when too many nodes are present in the network. With 120 nodes, only 80% of the packets arrive to the destination. OLSR presents a lower delivery ratio: the flooding of `Topology Packets` is required and create collisions. Besides, flooding being not reliable, some packets are lost, and sometimes no route is present in the routing table of the source. Thus, some packets are dropped. VSR presents the highest delivery ratio: the hierarchy allows to execute a different routing protocol in each hierarchy. This improves the scalability, and thus the delivery ratio. Moreover, the clusters form a stable topology: less route reconstructions occur, limiting the packet losses. The self-organization structure allows to optimize the per-

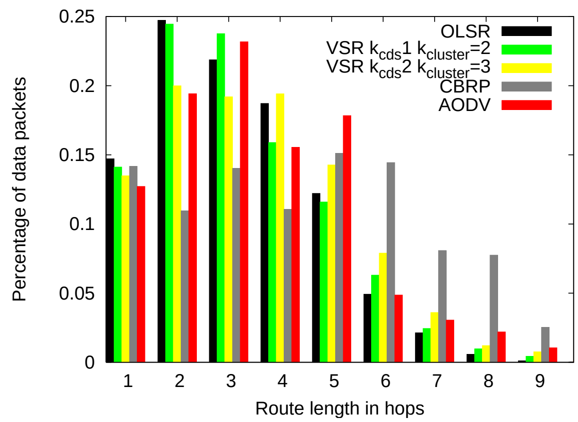

Figure 6. *The Route Length Distribution*

formance of classical routing protocols. Even with 120 nodes, more than 96% of the packets are efficiently delivered.

5.2. *Vertical Scalability*

OLSR and VSR present the lowest delay, invariant with the load of the network. The delay of VSR is a little smaller

When $k_{cds} = 2/k_{cluster} = 3$, the clusters comprise more nodes than when $k_{cds} = 1/k_{cluster} = 2$. Thus the ratio of intra-cluster routes is larger. This reduces the number of route discoverings. However, the route discovering delay seems very small, and the global delay is similar whatever the values of k_{cds} and $k_{cluster}$ are. The delay of AODV decreases when the number of simultaneous connections increases: more `RREQ` are sent across the network, creating more entries in the routing tables. The probability to have already the destination in its routing table increases, avoiding the delay of route discovering. CBRP suffers from the source routing: the routes are only cached in the source, reducing the probability to have a valid route before a route discovering. Moreover, the `RREQ` of CBRP are flooded along a redundant structure of clusters and gateways. Thus, the overhead is large, collisions occur, and the global delay increases. These remarks are corroborated with the results presented in figure 7. The latency of the route discovering of VSR is very scalable with the load. Moreover, VSR presents the lowest delay: the flooding along the backbone seems very efficient to reduce the overhead, and then the delay. The backbone is well-exploited. The latency of AODV increases slightly when the load increases (but the global delay decreases because the number of route discovering decreases). CBRP presents the highest delay to discover a route.

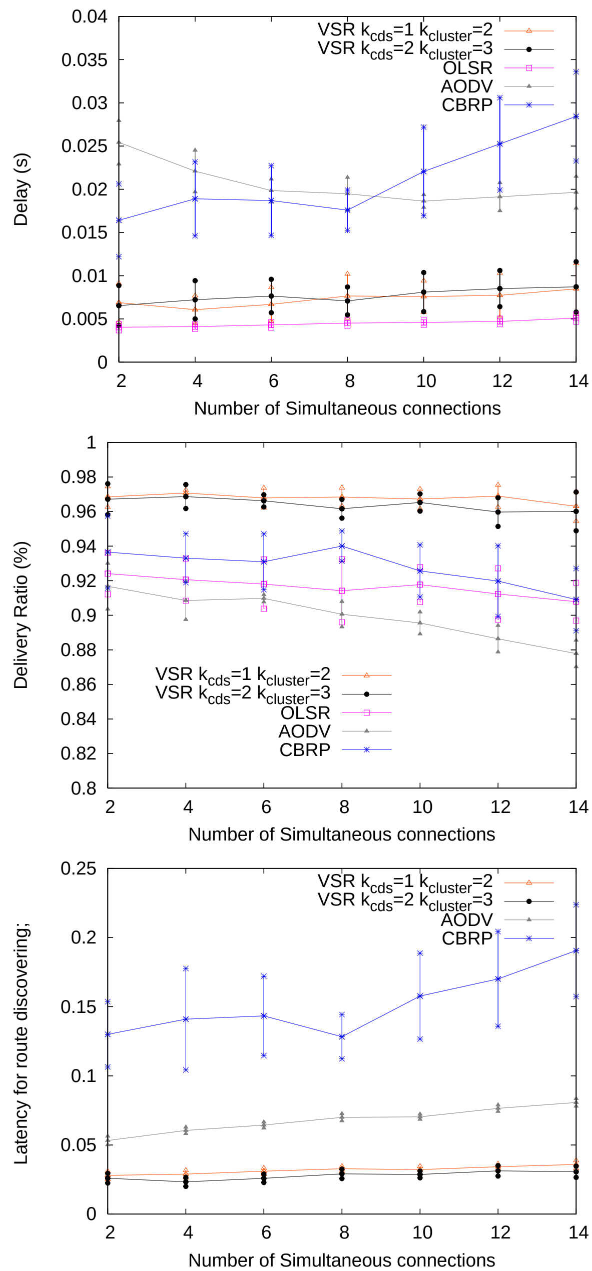

Figure 7. *Vertical scalability*

Finally, the delivery ratio is studied. CBRP presents the lowest delivery ratio because many route discovering are unsuccessful. AODV presents an higher delivery ratio but its performances decrease when the load increases: the overhead of the route discovering tends to overload the medium, but its impact is limited. The delivery ratio of OLSR and AODV seem to be comparable. VSR presents the highest delivery ratio since it constructs stable routes, and its route discovering is efficient, because of the virtual structure of self-organization.

5.3. *Mobility*

Since in MANet, all the nodes are mobile, we study the impact of such a mobility, varying the maximum speed of the random waypoint from 0 to 30 m.s^{-1}. However, a null speed can lead to a stationary state in which the whole network stops moving (Yoon *et al.,* 2003). Thus, a new speed is automatically chosen after a finite time when a node keeps the same speed during a too long time. The delay of OLSR does not change much when the mobility increases: proactive protocols allow to update periodically their knowledge of the network. Thus, no delay is required. The delay of VSR is also stable, the route discovering though the backbone being efficient. AODV presents an higher delay since a route discovering is required before having an available route. The delay of CBRP tends to increase when the mobility is large: routes are less robust and tend to break more often, requiring a supplementary delay to re-initiate the route discovering.

The same remark could be given about the delivery ratio: CBRP suffers from packet losses, the delivery ratio is the lowest among all the protocols and decreases quickly with topology changes. The delivery ratio for all protocols decreases when the mobility increases: more topology changes occur, creating routes breaks, and dropped packets. However, VSR keeps on presenting the highest delivery ratio because the route discovering passes uniquely through the backbone nodes and because VSR mixes efficiently the proactive and reactive behaviors because of the hierarchy.

5.4. *Density*

We can note that the delay and the packet losses decrease when the density increases (fig. 9). Indeed, the diameter of the network is smaller, and route are in consequence also shorter. When the network is of radius one (all routes are single hop), all the protocols seem to react well. However, for low densities, the reactive protocols present an higher delay since the route discovering is longer and more transmissions are required. In the same way, the delivery ratio of reactive protocols is smaller. The cause of this remark is perhaps the lack of reliability of broadcasts: some `RREQ` are lost because the network is very sparse, and the redundancy in the route discovering decreases. The delay of VSR increases for low density networks since the route discovering must be forwarded farther, but this delay remains very similar to the delay of OLSR. VSR keeps on presenting the lowest ratio of packet losses, whatever the degree

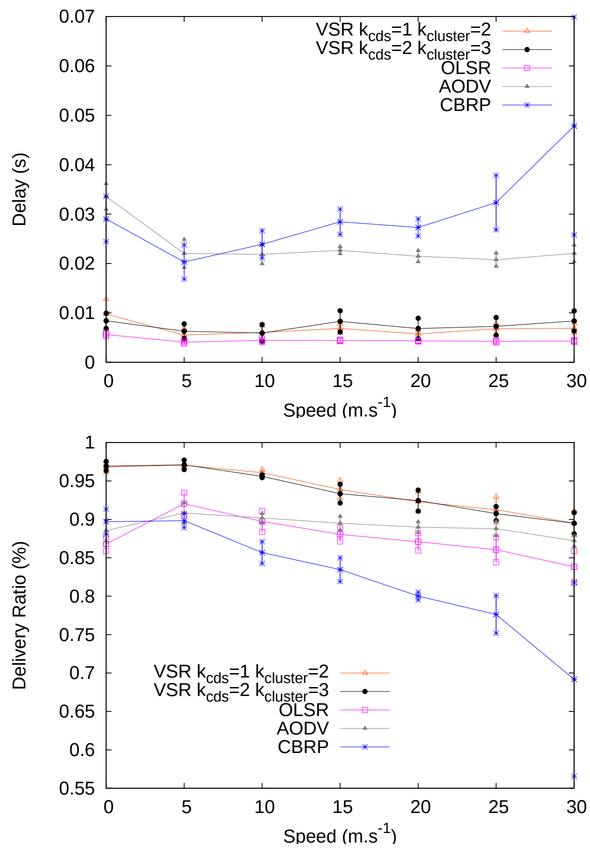

Figure 8. *Impact of the mobility*

is. Finally, the delay of VSR with k$_{cluster}$ >2 in very dense networks is the highest among all other protocols: `hello` packets must be forwarded by all the neighbors of the same clusters. Thus, when the diameter of the network reaches one hop, all the nodes must forward all the `hellos`. This yield an heavy loaded medium, increasing the delay. However, a topology control like (Li *et al.,* 2005) could be applied, reducing the degree, and then the overhead. In the same way, OLSR could be implemented as the proactive routing protocol inside a cluster.

5.5. *Overhead*

The overhead is investigated (tab. 1). We measured the number of packet per second per node. Firstly, we separated all the types of control packets to understand finely the source of the overhead of each protocol. VSR presents a proactive part but

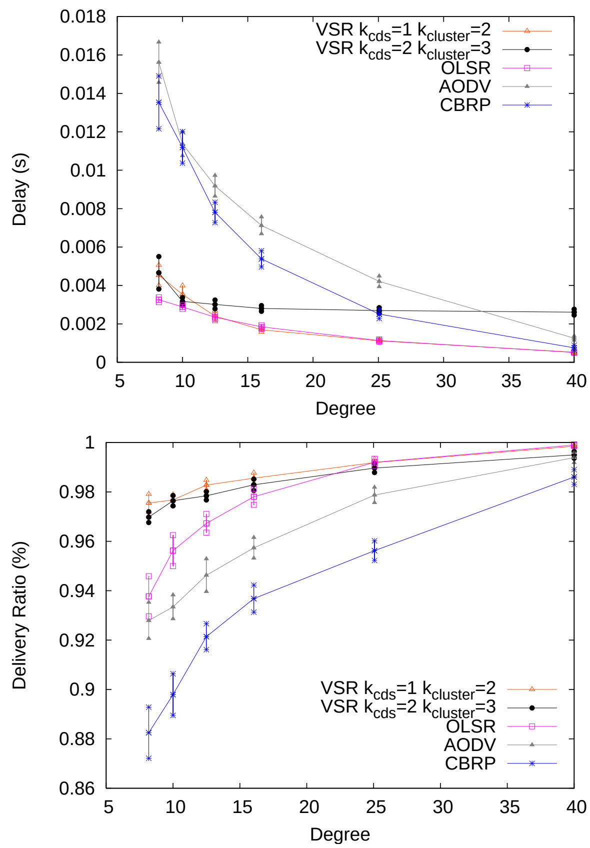

Figure 9. *Impact of the density*

this allows a great reduction of the reactive part: route discovering are only for inter-cluster routes and the overhead is optimized thanks to the backbone. In consequence, VSR presents the lowest overhead among all the protocols with k_{cds}=1 and k_{cds}=2: since each node must send all its neighbors in `hellos`, one transmission is sufficient for the backbone maintenance. The cluster topology knowledge being only partial, the overhead remains reasonable. However, when $k_{cluster}$=3, the overhead is too large: OLSR or a topology control algorithm must be applied to reduce the proactive overhead in order to avoid excessive packet collisions. Oppositely, the reactive part decreases when $k_{cluster}$ increases because less nodes relay the `RREQ` and because route discovering are less frequent. OLSR is a proactive protocol and requires the periodical flooding of `Topology Packets`. The overhead is in consequence significant: 1.4 packet per second are needed for the global topology knowledge. CBRP and AODV for a low traffic present a reasonable overhead. However, the route discovering of

	Proactive part			Reactive Part		Total
	Hellos	Topology Packets	Virtual Structure	RREQ	RREP	
VSR k_{cds}=1 $k_{cluster}$=2	0.25	N/A	0.26	0.06	0.028	0.59
VSR k_{cds}=2 $k_{cluster}$=3	3.1	N/A	0.27	0.009	0.005	3.4
OLSR	0.4	1.4	N/A	N/A	N/A	1.8
AODV	N/A	N/A	N/A	0.39	0.22	0.61
CBRP	0.49	N/A	N/A	0.9	0.09	0.99

Table 1. *Overheads in packet per second per node*

CBRP is less efficient since a `RREQ` can be forwarded several times by a gateway to different clusterheads. Thus the overhead of CBRP is larger.

Then, we investigated the impact of the load of the network on the overhead (fig. 10). The overhead of OLSR is stable: it does not depend on the traffic. The overhead of AODV and CBRP increases: more route discoverings are required. CBRP presents an higher overhead than AODV. Finally, VSR presents a very scalable overhead: the dissemination of `RREQ` through the backbone is efficient.

Then, we study the impact of the number of nodes. AODV is very scalable: when the networks carries a low traffic, AODV has a stable overhead: the number of `RREQ` does not increase since we maintain constant the number of connections. The overhead of CBRP, because of its flooding through the clusterheads and gateways, increases when more nodes are present. The overhead of OLSR increases: more nodes must send `Topology packets`. Besides, VSR presents a very scalable overhead, proposing a trade-off between hybrid and proactive approaches. VSR with k_{cds}=1 presents the lowest overhead. VSR is scalable both when we increase the number of nodes and when we increase the number of connections.

5.6. *Route repairs*

Finally, the impact of the route repair mechanisms is studied. To have the most general approach, we do not assume the existence of a cooperation among the MAC layer and the network layer. In the same way, the promiscuous mode is considered non available. An `Acknowledgment` must be sent by each node when it receives a packet for itself or to forward. If no `Ack` is received after 3 transmissions, the next hop is considered dead and a route repair is initiated. We compare the route repair mechanism efficiency of CBRP and VSR. Route repairs introduce timeout mechanisms and retransmissions. Thus, the delay is increased for both CBRP and VSR. However, since CBRP presents already an higher delay than VSR without route repairs, CBRP keeps on presenting an higher delay. Oppositely, the delivery ratio is greatly improved. VSR seems not suffer from the mobility : route repairs allow to maintain a 98% delivery ratio even at 30m.s^{-1}.

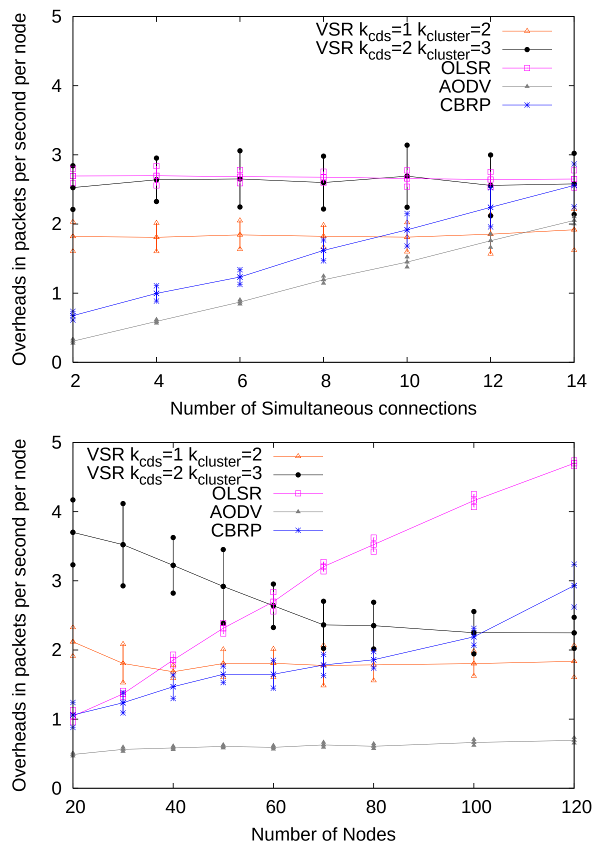

Figure 10. *Overheads of the different routing protocols*

6. Concluding remarks and Future Work

MANets seem promised to a large development thanks to their flexibility. However, many problems remain to be addressed like routing, addresses assignment, mobility management, Internet interconnection…Two main ideas are opposed to solve these issues: first, to consider a flat network where unicast routing protocols, mobility management are proposed; second, to introduce self-organization in order to structure the network using virtual topologies like backbone and clusters. Such a self-organization takes into account the nodes properties and heterogeneity, and try to limit the impact of the mobility. Self-organized topologies can be also viewed as a framework to develop network services like routing, mobility management,…

In this paper, we have proposed a routing protocol based on a virtual-structure of self-organization. The backbone nodes are chosen according to a stability weight.

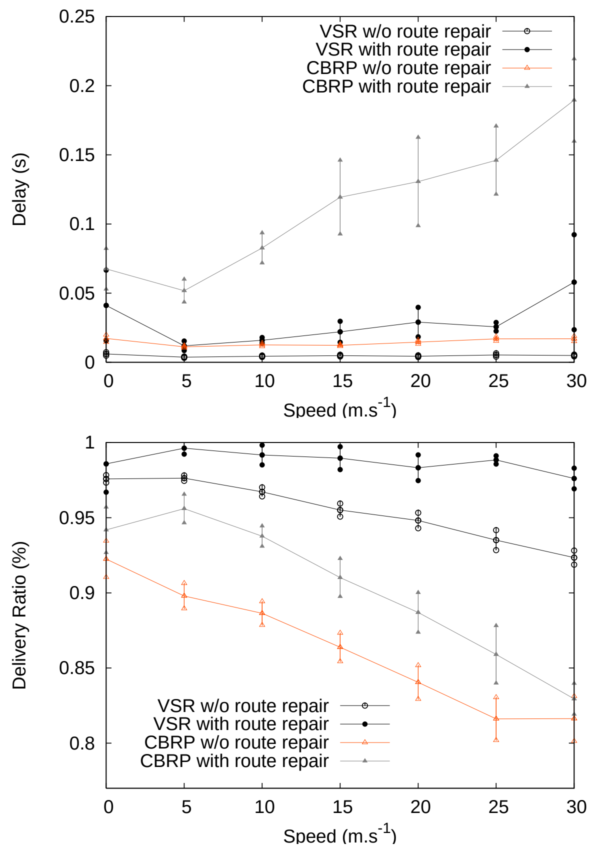

Figure 11. *Impact of the route repair mechanism of CBRP and VSR*

Thus, backbone clients being weaker nodes can delegate control procedures to their parent in the backbone. Moreover, the backbone helps to resolve the broadcast storm: only a subset of the nodes is allowed to forward the control packets, reducing the load, and allowing to stop quickly the flooding process in the network. Clusters are equally exploited: they help to hierarchize the network and propose a stable view of the topology. This hierarchy allows to execute different routing protocols in local and in distant, improving the scalability. VSR is a routing framework constituted by a proactive protocol inside a cluster, and by a reactive routing protocol in inter-cluster. Any routing protocol can be adapted to VSR. The virtual structure of self-organization improves greatly the performances of classical routing protocols in exploiting a stable virtual topology, reducing the broadcast storm problem and limiting the overhead. VSR provides a trade-off between proactive approach and reactive one.

As a future work, we plan to extend VSR, so that the virtual structure of self-organization is fully-exploited: a multicast extension could be proposed. In the same way a mobility management protocol optimizing the interconnection between the ad hoc area and the Internet, with an optimized integration with Mobile IP could be investigated. Additionally, testbed experiments must be conducted in order to test the performances in real environments.

7. References

Alzoubi K. M., Wan P.-J., Frieder O., "Distributed Heuristics for Connected Dominating Set in Wireless Ad Hoc Networks" , *IEEE ComSoc/KICS Journal of Communications and Networks, Special Issue on Innovations in Ad Hoc Mobile Pervasive Networks*, vol. 4,1, p. 22-29, march, 2002.

Amis A., Prakash R., Vuong T., Huynh D., "Max-min d-cluster formation in wireless ad hoc networks" , *INFOCOM*, IEEE, Tel-Aviv, Israel, p. 32-41, March, 1999.

Boukerche A., "Performance Evaluation of Routing Protocols for Ad Hoc Wireless Networks" , *Mobile Networks and Applications*, vol. 9,4, p. 333-342, August, 2004.

Butenko S., Cheng X., Du D.-Z., Pardalos P. M., "On the construction of virtual backbone for ad hoc wireless networks" , *Cooperative Control: Models, Applications and Algorithms*, vol. 1 of *Cooperative Systems*, Boston, Kluwer Academic Publishers, chapter 3, p. 43-54, January, 2003.

Clausen T., Jacquet P., Optimized Link State Routing Protocol (OLSR), RFC 3626, IETF, October, 2003.

Feeney L., Nilson M., "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment" , *INFOCOM*, IEEE, Anchorage, USA, p. 1548-1557, April, 2001.

Haas Z. J., Pearlman M. R., "The Performance of Query Control Schemes for the Zone Routing Protocol" , *SIGCOMM*, ACM, Vancouver, Canada, September, 1998.

Helmy A., "Contact-extended Zone-based Routing for Transactions in Ad Hoc Networks" , *IEEE Transactions on Vehicular Technology*, vol. 54,1, p. 307-319, January, 2005.

Jiang M., Li J., Tay Y. C., Cluster Based Routing Protocol (CBRP), Internet draft version 01, IETF, July, 1999.

Johnson D. B., Maltz D. A., Hu Y.-C., The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), Internet draft version 10, IETF, July, 2004.

Li J., Blake C., de Decouto D. S. J., Lee H. I., Morris R., "Capacity of Ad Hoc Wireless Networks" , *International Conference on Mobile Computing and Networking (MOBICOM)*, ACM, Roma, Italy, July, 2001.

Li x.-Y., Song W.-Z., Wang Y., "Efficient Topology Control for Ad-hoc Wireless Networks with Non-uniform Transmission Ranges" , *Wireless Networks*, vol. 11,3, p. 255-264, May, 2005.

Lin C. R., Gerla M., "Adaptive Clustering for Mobile Wireless Networks" , *IEEE Journal of Selected Areas in Communications*, vol. 15,7, p. 1265-1275, 1997.

Mario G., "From battlefields to urban grids: New research challenges in ad hoc wireless networks" , *Pervasive and mobile computing*, vol. 1,1, p. 77-93, March, 2005.

Ni S.-Y., Tseng Y.-C., Chen Y.-S., Sheu J.-P., "The Broadcast Storm Problem in a Mobile Ad hoc Network" , *International Conference on Mobile Computing and Networking (MOBICOM)*, ACM, Seattle, USA, p. 151-162, August, 1999.

OPNET Modeler, "`http://www.opnet.com` (v8.1a), Bethesda, MD" , n.d.

Pearlman M. R., Haas Z. J., "Determining the Optimal Configuration of the Zone Routing Protocol" , *IEEE Journal on Selected Areas in Communications*, vol. 17,8, p. 1395-1414, June, 1999.

Perkins C. E., "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers" , *SIGCOMM*, ACM, London, United Kingdom, p. 234-244, August, 1994.

Perkins C. E., Belding Royer E. M., Das S. R., Ad hoc On-Demand Distance Vector (AODV) Routing, RFC 3561, IETF, July, 2003.

Sinha P., Sivakumar R., Bharghavan V., "CEDAR: a Core-Extraction Distributed Ad Hoc Routing Algorithm" , *INFOCOM*, IEEE, New York, USA, p. 202-209, March, 1999.

Sivakumar R., Das B., Bharghavan V., "Spine routing in ad hoc networks" , *Cluster Computing*, vol. 1,2, p. 237-248, 1998.

Theoleyre F., Valois F., "About the Self-Stabilization of a Virtual Topology for Self-Organization in Ad Hoc Networks" , *in* , LNCS (ed.), *Self-Stabilization Symposium (SSS)*, vol. 3764, IEEE, Barcelona, Spain, p. 214-228, October, 2005.

Theoleyre F., Valois F., "A Self-Organization Structure for Hybrid Networks" , *Ad Hoc Networks*, 2007. to appear.

Wu J., Dai F., "Distributed Dominant Pruning in ad hoc wireless networks" , *International Conference on Communications (ICC)*, IEEE, Anchorage, USA, p. 353-357, May, 2003.

Wu J., Li H., "Dominating-set-based routing in ad hoc wireless networks" , *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL'M)*, ACM, Seattle, USA, p. 7-14, August, 1999.

Wu J., Lou W., "Extended Multipoint Relays to Determine Connected Dominating Sets in MANETS" , *Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, IEEE, Santa Clara, USA, October, 2004.

Yoon J., Liu M., Noble B., "Random Waypoint Considered Harmful" , *INFOCOM*, IEEE, San Francisco, USA, March, 2003.

Fabrice Theoleyre *Fabrice Theoleyre obtained a M.S. degree in Electrical and Computer Engineering (2003) and a Ph.D. in Computer Science (2006) both from INSA of Lyon, France. He was also a postdoctoral fellow in the LSR (Grenoble) and an associate professor in the LIRIS (INSA of Lyon). Since november 2007, he is a CNRS research scientist (charge de recherche) in the Grenoble Informatics Laboratory (LIG). His research interests are generally in ad hoc and mesh networks, including distributed algorithms, self-organization, testbeds, MAC layer and geographical routing.*

Fabrice Valois *Fabrice Valois obtained a Master of Sciences in 1995 in Computer Science (networking and performance evaluation) and a Ph.D. in january 2000 about Performance Evaluation of Hierarchical Cellular Networks, both from the PRiSM laboratory, University of Versailles, France. Since September 2000, Fabrice Valois is associate professor in the Telecom-*

munications Department of INSA Lyon, France. He is also a member of the CITI laboratory and the INRIA ARES project. His research interests are in the area of mobile ad hoc networks, wireless sensor networks and mesh networks. More particularly, his research works are focused on self-organisation, routing & broadcasting strategies for MANET and WSN. He has also interests about performance analysis of MAC layer for MANET using a Markov Process Algebra (PEPA) and simulation tools.